MANUAL

Database Metadata Receiver

5/2017

[1 Introduction 3](#__RefHeading___Toc613_1629593180)

[1.1 The Program 3](#__RefHeading___Toc808_1602930733)

[1.2 Recommended hardware requirements: 4](#__RefHeading___Toc420_2008615432)

[1.3 Software requirements: 4](#__RefHeading___Toc422_2008615432)

[1.4 Software specific information & Web Page 4](#__RefHeading___Toc826_1214427482)

[2 Config 5](#__RefHeading___Toc417_2008615432)

[2.1 Default-Config.txt : 6](#__RefHeading___Toc840_1602930733)

[2.2 Databases.txt 7](#__RefHeading___Toc745_639229776)

[3 SQL 8](#__RefHeading___Toc747_639229776)

[3.1 SQL-Default values 8](#__RefHeading___Toc749_639229776)

[3.2 SQL Tutorial with screenshots 8](#__RefHeading___Toc615_1629593180)

[3.2.1 Selecting the database type 8](#__RefHeading___Toc617_1629593180)

[3.2.2 Entering the connection and file information 9](#__RefHeading___Toc619_1629593180)

[3.2.3 Selecting databases 10](#__RefHeading___Toc621_1629593180)

[3.2.4 Data processing 11](#__RefHeading___Toc623_1629593180)

[3.2.5 Finishing 11](#__RefHeading___Toc625_1629593180)

[4 Firebird 12](#__RefHeading___Toc751_639229776)

[4.1 Firebird-Default values 12](#__RefHeading___Toc753_639229776)

[4.2 Firebird Tutorial with screenshots 12](#__RefHeading___Toc627_1629593180)

[4.2.1 Selecting the database type 12](#__RefHeading___Toc629_1629593180)

[4.2.2 Entering the connection and file information 13](#__RefHeading___Toc631_1629593180)

[4.2.3 Data processing 14](#__RefHeading___Toc633_1629593180)

[4.2.4 Finishing 14](#__RefHeading___Toc635_1629593180)

# Introduction

This program is used for receiving Metadata structure from a database,

The received data includes the db Name, tables inside the db, columns inside the tables, And datatype for each column.

The result is then written to a file, of which the: Path, name and filetype can be edited independently.

The program also includes **2** .txt Files:

- First one being the **Config**, which can be used to autofill information for the program, instead of filling the required information during each startup.

- Second one is a **list of databases**, which the config may use, if such option is selected.

Both of these mentioned files, will be created during startup, if they’re not found on the same directory, as where the program is located.

On the github page, a Namelister.exe is also provided.  
With the Namelister, you can create a .txt file from every existing file, from the same location.   
>> Place the namelister in the folder you have your databases  
>> Run it once  
>> You will get a list of DB names  
>> You can rename the list to Databases.txt / copy the contents to Databases.txt

>> Now you have a list of all databases from the folder

## The Program

The program itself works with a principle of:

1. Insert User Information (User Id, Password, etc.)
2. Try connection to server → [Fail] → Back to 1
3. Construct a connectionstring from entered details
4. For loop for each selected database
5. For loop for each table inside database
6. For loop for each column inside table
7. Create lists from all columns

( File = File + Databasename[i] + ";" + Tablename[I]+ ";" + Columnname[N] + ";" + Columntype[N] + Environment.NewLine; )  
[**ExampleDatabase;ExampleTable;ExampleColumn;ExampleColumnType**]

1. Write the file, and ask to open it.  
   (System.IO.File.WriteAllText (Path, File); )

## **Recommended hardware requirements:**

2 GHZ Dual core processor or better  
2 GB System memory  
1 GB of free HDD space\*  
TCP/IP access to the database, → LAN Connection  
\*May vary depending on the size of database  
\*eg. 100 Tables, 10 columns each, produces a file of approx: 50KB, in Size

## Software requirements:

-- Possibly system admin rights / Super User rights  
-- Windows based OS  
-- Windows Vista or higher  
-- .Net Framework 4.0 or 4.5  
  
-- Linux based OS  
-- EG. Ubuntu 16.04   
–- Mono - (.Net Framework for Linux)  
-- .DLL files included with the program

Windows version:  
Linux version:   
To be noted:  
The Linux version was tested on a Windows Server, and had no problems at all in case of running properly.

## Software specific information & Web Page

The software is created using Visualstudio, Monodevelop, C# Language.

Devs:

* Ville Kokkarinen | Git: <https://github.com/VilleKokkarinen/>  
  Ville Kokkarinen | Web: <https://villekokkarinen.github.io/>
* Aki Kaukonen | Git: <https://github.com/KiKizKi/>

This software is under the MIT license:  
Copyright (c) <2017> <Ville Kokkarinen> <Aki Kaukonen>

**Permission** is hereby **granted**, **free of charge**, to **any person** obtaining

a copy of this software and associated documentation files.  
  
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND

# **Config**

Config.txt is located in the **same folder** as the application.exe  
  
**If** Config.txt cannot be found when application is started, it **will** create a new blank one.

Config **is not used by default**, the user has to modify the:  
**line #3** in the config file.

By enabling the use of Config you can preset a variety of settings.

**Settings:**

* **UseConfig**= True/False   
  – Defines whether to use the config at all.
* **SQL**= True/False  
   – Defines which databasetype using. True = SQL, False = FB.
* **Path**=/home/user/Desktop   
  – Path where to save the data.
* **Filename**=File.csv   
  – Name and datatype for the file.
* **UsePremadeConnectionString**=True/False  
  – Defines whether to use a premade C-S.
* **Connectionstring**="User ID=[User];Password=[P-W];Server=[Server-IP];"  
  - The Connectionstring which to use. - \*NOTE: When using firebird, AND Databases is set to True. You MUST provide a working databasename in the connectionstring.
* **Databases**=True/False  
  – Defines whether to use databases found in databases.txt
* **MultipleFiles**=True/False

– Defines whether to create 1 massive file, or a new file for each database.  
– Also changes the Filename as the databasename for each file.

* **AddDates**=True/False   
  – Defines whether to add current System Date to filename.  
  – Works with Multiplefiles.

## Default-Config.txt :

﻿config file, Edit each Line after the character "=",REMEMBER NOT TO add a space between the question and the "="

//Example of a method filled the right way: USECONFIG=TRUE

//true=Uses this Config file

//false=doesn't

UseConfig=false

//true=uses SQL

//false=uses Firebird

SQL=true

//Path for the file [Default: /home/[Your\_Username]/Documents]

//Name for the file

Path=/home/user/Desktop

Filename=File.csv

//true = uses premade connection defined below [SQL : "User ID=sa;Password=admin;Server=192.168.0.0;"]

//false = asks required information on startup [FBD : "User ID=sysdba;Password=admin;Database=192.168.0.0:X:/Folder/DBname.fdb;DataSource=192.168.0.0;"]

UsePremadeConnectionString=true

Connectionstring="User ID=sa;Password=admin;Server=192.168.0.0;"

//true= uses databasenames found from the "Databases.txt" file

//false= asks names on startup

//\*ALL\* Chooses All found \*SQL Databases from the connection

Databases=false

// ^ ! Each database on a new line ! ^

//true Creates a different file for every database \*Renames the file as the database. Regardless of option above

//false Creates 1 big file, with all databases in it

MultipleFiles=true

//true Adds the current system date (DD/MM/YY), to the filename, eg. (DB\_04-05-17)

//false Doesn't

AddDates=true

//IF you've lost your [Databases.txt] file, Just manually create one at the same directory where THIS config.txt is in

//Name it exactly: Databases.txt

## Databases.txt

Databases.txt, is located in the same location as the Application.exe

Databases.txt, determines the databases that the application will receive metadata from.  
To select multiple Firebird databases, one must use the databases.txt.  
Using SQL it’s not necessary.

If Databases.txt cannot be found when application is started it will be created to the default location (Application.exe root folder).

Fill in the databases.txt as shown here:  
Or use the Namelister.exe Mentioned on page 1.  
  
*database1  
database2*

*database3*

*…*

# SQL

## SQL-Default values

The mark **\*\*** means that you can leave the line empty and the program will use the default value.

Default values on SQL:

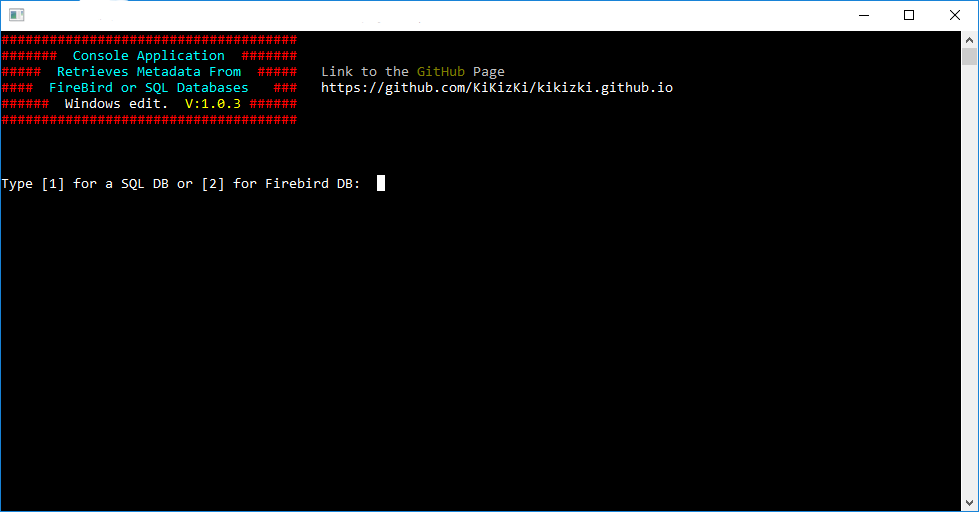
User ID: sa

File name: File.csv

Path: current directory of the program

## SQL Tutorial with screenshots

### Selecting the database type

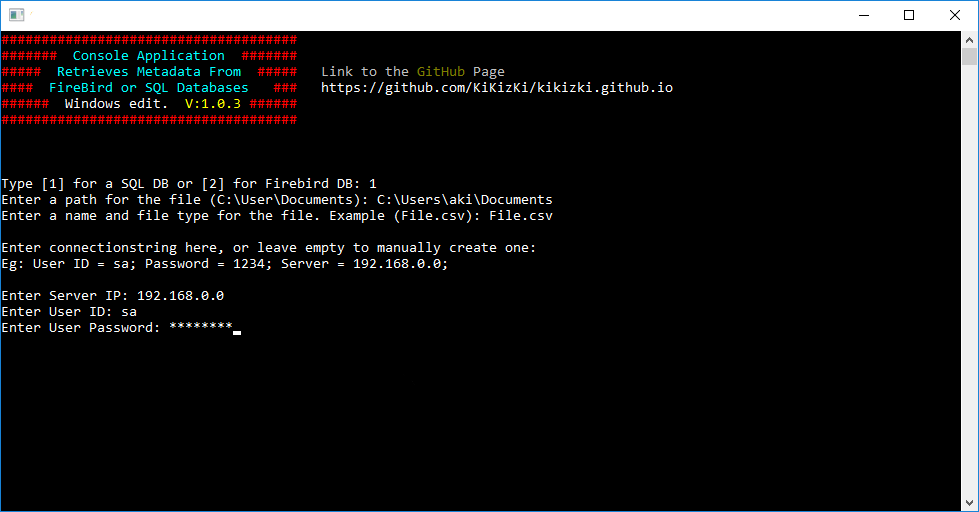


>Type **1** or **2** to select the database type.   
 >Then press Enter to confirm.

[1] = SQL

[2] = Firebird

### Entering the connection and file information



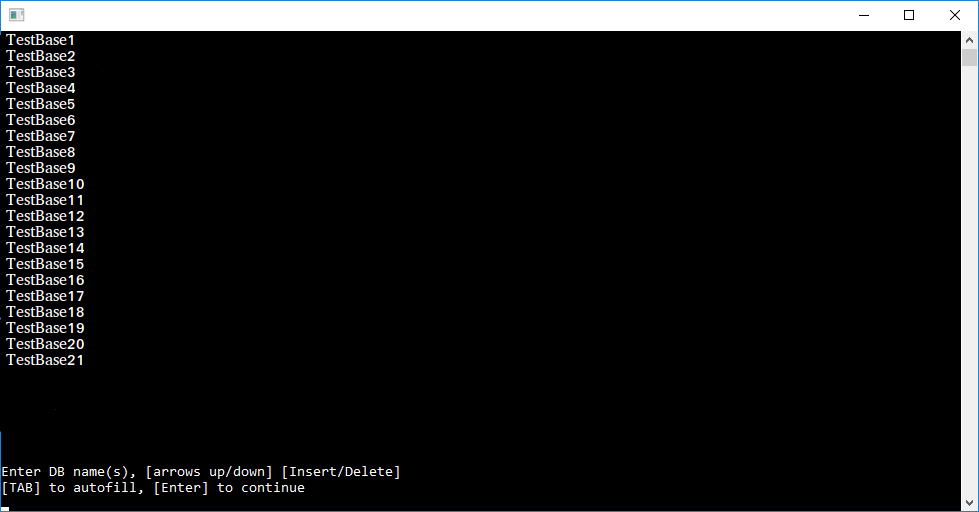
> Enter a path for the file, *or leave empty to use the default.*  
> Enter the name, and type for the file, *or leave empty to use the default.*

> Enter The login credentials for the server:  
 > Server IP  
 > User ID

> User Password,   
 *Password is typed hidden from view, and shows Stars as symbols (\*).*

Press Enter after each selection to confirm your selection.

### Selecting databases



The program lists databases that are accessible from the selected SQL server.

>You can type the databases you want to connect to,

[TAB] key will autofill the closest match from the databaselist.

>You can browse the list by using **[up] & [down]** arrow keys.

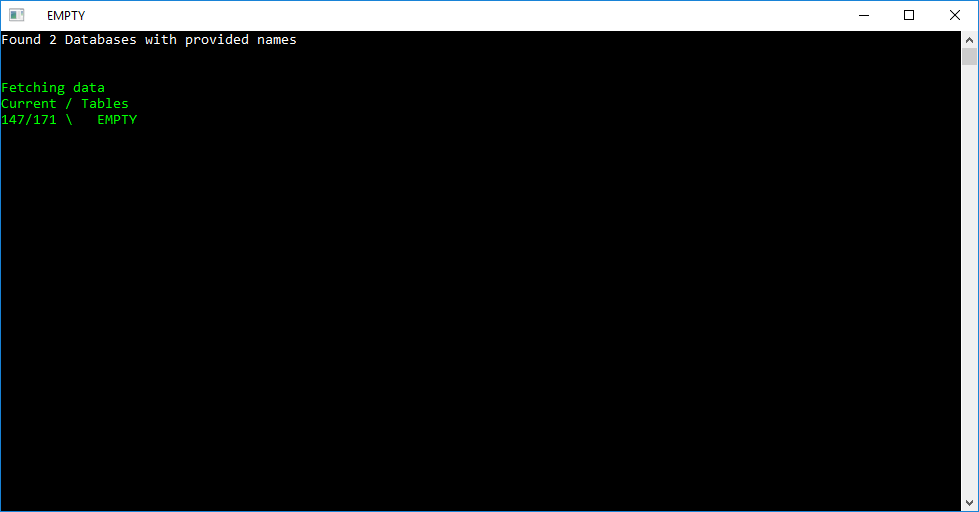
>Add the wanted databases to the connection list by pressing the **[insert]** key while a database is selected.

>You can delete databases from the connection list by using the **[delete]** key.

Pressing the **[delete]** key will remove the last added item from the list.

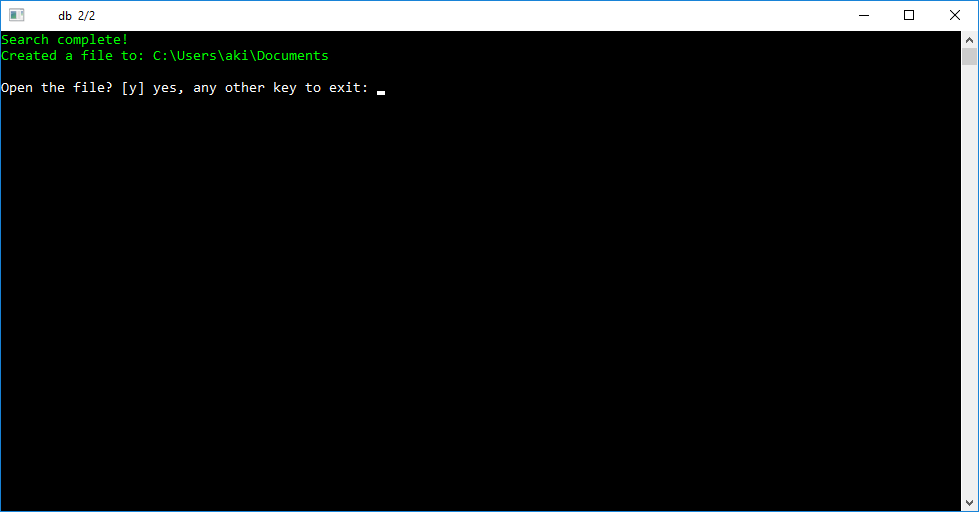
Press **Enter** to continue with the program, and start receiving data.

### Data processing



The program starts reading data from the database, and creates a file from the data.

### Finishing



The program will write the file, and possibly ask to open it.

# Firebird

## Firebird-Default values

The mark **\*\*** means that you can leave the line empty and the program will use the default value.  
Default values on Firebird:

User ID: sysdba

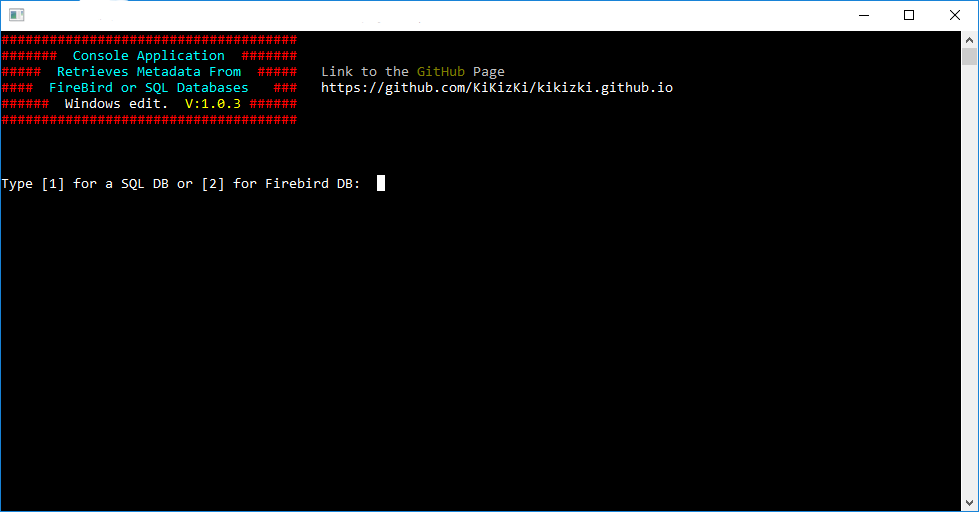
File name: File.csv

Path: current directory of the program

Port: 3050

## Firebird Tutorial with screenshots

### Selecting the database type

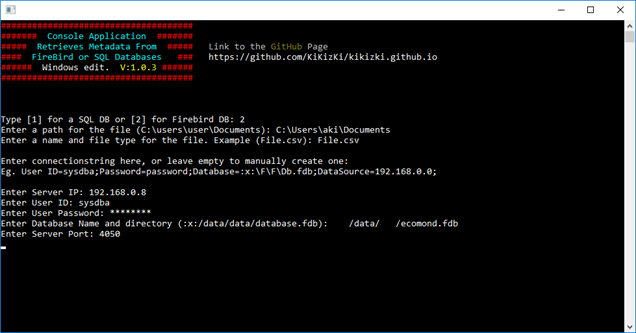


>Type **1** or **2** to select the database type.   
 >Then press Enter to confirm.

[1] = SQL

[2] = Firebird

### Entering the connection and file information



> Enter a path for the file, *or leave empty to use the default.*  
> Enter the name, and type for the file, *or leave empty to use the default.*

> Enter The login credentials for the server:  
 > Server IP  
 > User ID

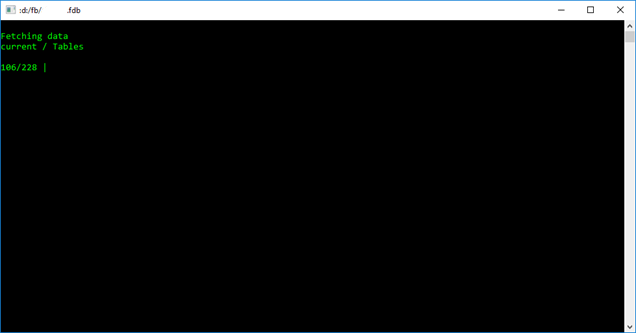
> User Password,   
 *Password is typed hidden from view, and shows stars as symbols (\*).*

>Databasename and location.

>Port for the server, Default is 3050.

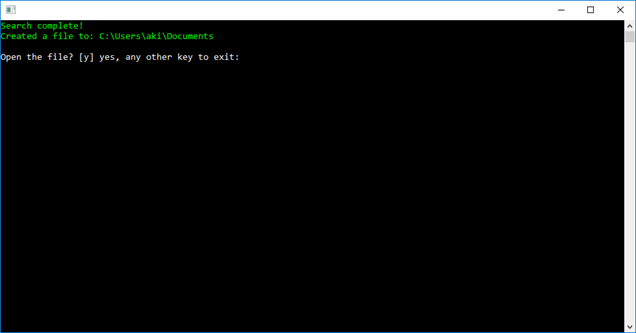
Press Enter after each selection to confirm your selection.

### Data processing



The program starts reading data from the database, and creates a file from the data.

### Finishing



The program will write the file, and possibly ask to open it.